US-PAT-NO: 6560647

DOCUMENT-IDENTIFIER: US 6560647 B1

TITLE: Enterprise management system and method which includes semantically correct summarization

---------- KWIC ---------

Application Filing Date - AD (1):
19990304

Brief Summary Text - BSTX (5):
The data processing resources of-business organizations are increasingly taking the form of a distributed computing environment in which data and processing are dispersed over a network comprising many interconnected, heterogeneous, geographically remote computers. Such a computing environment is commonly referred to as an enterprise computing environment, or simply an enterprise. Managers of the enterprise often employ software packages known as enterprise management systems to **monitor,** analyze, and manage the **resources** of the enterprise. Enterprise management systems may provide for the collection of measurements, or metrics, concerning the resources of individual systems. For example, an enterprise management system might include a software agent on an individual computer system for the **monitoring** of particular **resources** such as CPU usage or disk access. The enterprise management agent might periodically collect metric data and write to a "data spill" containing historical metric data, i.e., metric data previously collected over a period of time. U.S. Pat. No. 5,655,081 discloses one example of an enterprise management system.

Detailed Description Text - DETX (2):
U.S. Pat. No. 5,655,081 titled "System for **Monitoring** and Managing Computer **Resources** and Applications Across a Distributed Environment Using an Intelligent Autonomous Agent Architecture" is hereby incorporated by reference as though fully and completely set forth herein.

Detailed Description Text - DETX (9):
When the computer programs are executed on one or more computer systems 150, an enterprise management system 180 is operable to **monitor,** analyze, and manage the computer programs, processes, and **resources** of the enterprise 100. Each computer system 150 in the enterprise 100 executes or runs a plurality of software applications or processes. Each software application or process consumes a portion of the resources of a computer system and/or network: for example, CPU time, system memory such as RAM, nonvolatile memory such as a hard disk, network bandwidth, and input/output (I/O). The enterprise management system 180 permits users to **monitor,** analyze, and manage **resource** usage on

heterogeneous computer systems 150 across the enterprise 100.

Detailed Description Text - DETX (13):
FIG. 4 shows an overview of the Monitor component 402 of the console node 400 of the enterprise management system 180. The Monitor 402 comprises a Manager Daemon 430, one or more Monitor Consoles (as illustrated, 420a and 420b), and a **Policy** Registration Queue 440. Although two Monitor Consoles 420a and 420b are shown in FIG. 4, the present invention contemplates that one or more Monitor Consoles may be executing on any of one or more console nodes 400.

Detailed Description Text - DETX (14):
In the preferred embodiment, the Monitor Consoles 420a and 420b use a graphical user interface (GUI) for user input and information display. Preferably, the Monitor Consoles 420a and 420b are capable of sending several different types of requests to an Agent 302, including: alert requests, update requests, graph requests, and drilldown requests. An alert request specifies one or more **thresholds** to be checked on a routine basis by the Agent 302 to detect a problem on the agent node 300. For example, an alert request might ask the Agent 302 to report to the Monitor Console 420a whenever usage of a particular software process exceeds a particular **threshold** relative to overall CPU usage on the agent node 300. An update request is a request for the status of the Agent 302. For example, the requested status information might include the version number of the Agent 302 or the presence of any alarms in the Agent 302. A graph request is a request to receive graph data, i.e., data on a metric as routinely collected by the Agent 302, and to receive the data in real time, i.e., whenever it becomes available from the present time onward. By obtaining and displaying graph data, the Monitor Console 420a enables the rapid identification and communication of potential application and system performance problems. Preferably, the Monitor Console 420a displays graph data in a graphical format. A drilldown request is a request to receive drilldown data, i.e., data on an entire metric group (a set of metrics) as collected by the Agent 302. By obtaining and displaying drilldown data, the Monitor Console 420a provides the ability to focus, in real-time, on a specific set of processes, sessions, or users. Preferably, the Monitor Console 420a displays drilldown data in a tabular format.

Detailed Description Text - DETX (15):
Whenever the Agent 302 generates an alarm to indicate a troublesome status on the agent node 300, the Manager Daemon 430 intercepts the alarm and feeds the alarm to one or more Monitor Consoles, such as 420a and 420b. Typically, an alarm is a notification that a particular **threshold** has been exceeded on a monitored process or subsystem on an agent node 300. The Manager Daemon 430 is capable of receiving alarms from a plurality of Agents 302. A Manager Daemon 430 is preferably always running on each console node 400 so that alarms can be captured even when the Monitor Consoles 420a and 420b are offline.

Detailed Description Text - DETX (16):
Each of the Monitor Consoles 420a and 420b is operable to issue one or more **policies**. A **policy** defines a disparate set of metrics to be collected on one

or more agent nodes 300. In other words, a **policy** allows a Monitor Console 420a or 420b to monitor one or more metrics on one or more agent nodes 300 simultaneously. For example, a user could build and deploy a **policy** that restricts web browser access on a plurality of agent nodes 300 with the following set of interrelated conditions: "IF more than 80% of server CPU is required by critical production applications, AND the run queue length is greater than six, AND active time on production disks exceeds 40%." **Policies** are registered with the **Policy** Registration Queue 440, from which they are disseminated to the appropriate Agents 302. An Agent 302 can execute a plurality of **policies** simultaneously.

Detailed Description Text - DETX (23):

FIG. 6 illustrates an overview of the Analyze component 406 of the console node 400 of the enterprise management system 180. In the preferred embodiment, Analyze 406 comprises the "ANALYZE" portion of the "BEST/1 FOR DISTRIBUTED SYSTEMS" software package available from BMC Software, Inc. Essentially, Analyze 406 takes the data collected by one or more Agents 302 and creates a model of one or more computer systems and the processes that run on those computer systems. In the preferred embodiment, Analyze 106 can model multi-vendor environments, system memory, multiple processors, disk drives, logical volumes, RAID devices, load **balancing,** ASCII and X terminals, local and remote file servers, independent and dependent transactions, client/server workloads, private and shared memory/transaction, CPU priority scheduling, networks of different types, and "ORACLE", "SYBASE", and "INFORMIX" database environments. In the preferred embodiment, Analyze 406 takes as input a domain file 466 which identifies the agent nodes 300 on the network and the relationship between them. Analyze 406 also takes as input a data repository in either UDF 212c or UDR 210c format, wherein the data repository 212c or 210c is a set of metric groups collected from one or more agent nodes 300.

Detailed Description Text - DETX (26):

FIG. 7 shows an overview of the Predict component 408 of the console node 400 of the enterprise management system 180. In the preferred embodiment, Predict 408 comprises the "BEST/1-PREDICT" component of the "BEST/1 FOR DISTRIBUTED SYSTEMS" software package available from BMC Software, Inc. Predict 408 is a planning tool which forecasts the impact of hypothetical changes on elements of the enterprise 100 such as disparate hardware, software, applications, and databases. Predict 408 takes the workload data from a Model File 468c, such as the Model File 468a generated by Analyze 406, and computes performance statistics such as workload response times, **utilization,** and throughputs at CPUs, disks, networks, and other elements of the enterprise computing environment 100. Thus, Predict 408 constructs a baseline model from collected data that represents the essence of the system under management. The user can also operate Predict 408 to construct the baseline model from pre-built model components, or from a combination of collected data and pre-built components. Preferably, Predict 408 uses a graphical user interface (GUI) for user input and information display.

Detailed Description Text - DETX (31):

Preferably, all of the metrics can be classified as one of a limited number

of key data types. Examples of data types include, but are not limited to, a counter, a gauge, or a string. A gauge is a number that can go up or down from data point to data point. For example, the speed of an automobile or the **utilization** percentage of a CPU would be measured by a gauge. A counter is a number that is monotonically increasing from one data point to the next: it can only go up, never down. For example, the odometer in an automobile, i.e., the indicator of total distance traveled, or the total number of disk accesses over the lifetime of a disk would be measured by a counter. A string is a series of characters which are manipulated as a group, as is well known in the art. Furthermore, the key data types may have additional variants, such as a clock, which is a form of counter representing elapsed time. Thus, each of the metrics has a semantic, i.e., a meaning, attached to it. The summarization method functions more intelligently by applying different summarization rules according to the semantics of the data type of each metric.